

ModelKeeper – Accelerating DNN Training via Automated Training Warmup

Authors: Fan Lai, Yinwei Dai, Harsha V. Madhyastha, Mosharaf Chowdhury

Institution: University of Michigan

Conference: 20th USENIX Symposium on Networked Systems Design and Implementation (NSDI), 2023

概述

该论文介绍了 **ModelKeeper**，一个自动化系统，旨在通过在共享机器学习集群中重用和转换先前训练模型的权重来加速深度神经网络（DNN）的训练。**ModelKeeper** 旨在利用模型之间的结构相似性，为新的训练任务提供有效的预热权重，从而减少 DNN 的训练时间。该方法具有可扩展性，可应用于各种任务、框架和模型架构。

动机和问题陈述

由于广泛的模型实验和部署，对深度神经网络（DNN）训练的需求不断增加，已在大型机器学习集群中形成瓶颈。尽管已有方法通过并行化或更好的资源管理来优化训练速度，但这些方法忽视了利用先前训练模型来初始化新训练任务的潜力。**ModelKeeper** 通过识别相似的预训练模型并转换其权重用于新模型，解决了这一问题，加速了收敛过程，降低了总体训练成本。

动机 1：广泛地模型相似度

首先，机器学习开发人员需要使用神经架构搜索（NAS）来选择最合适的模型架构。其次，开发人员可以训练多个模型，以定制延迟-准确性之间的权衡，或通过集成学习获得更好的预测效果。第三，随着用户数量的增加，相似模型的可能性也会增加。从 figure2 中的 top1 曲线可以看出在三个不同地点 workload 上，超过 60% 的模型都至少可以找到一个模型与它相似度超过 60%。

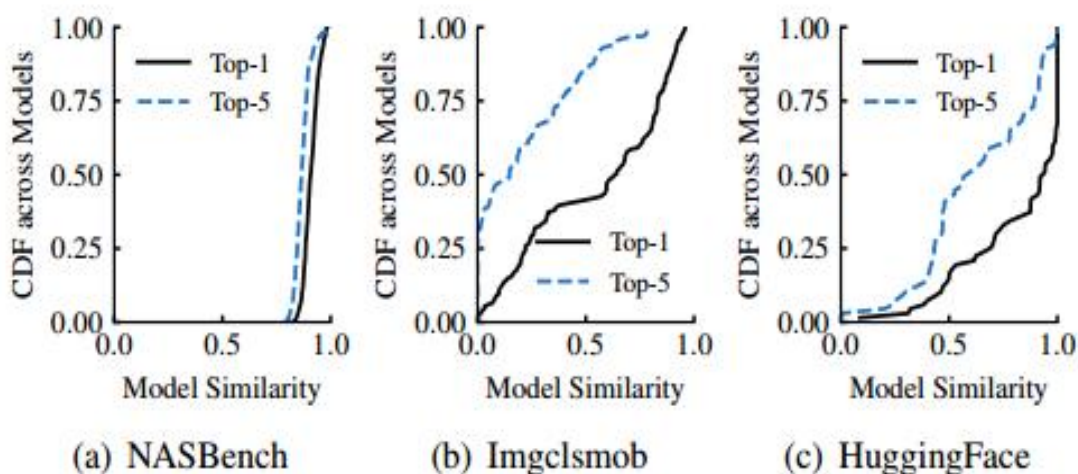
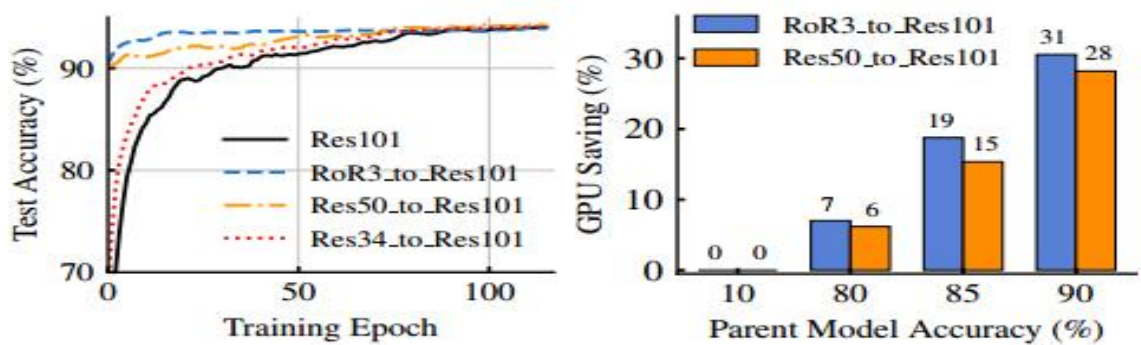


Figure 2: Pervasive model similarity in today’s model zoos. We measure the top-1 and top-5 architectural similarities of each model to other models, and report the distribution across models. 1 indicates identical model architectures.

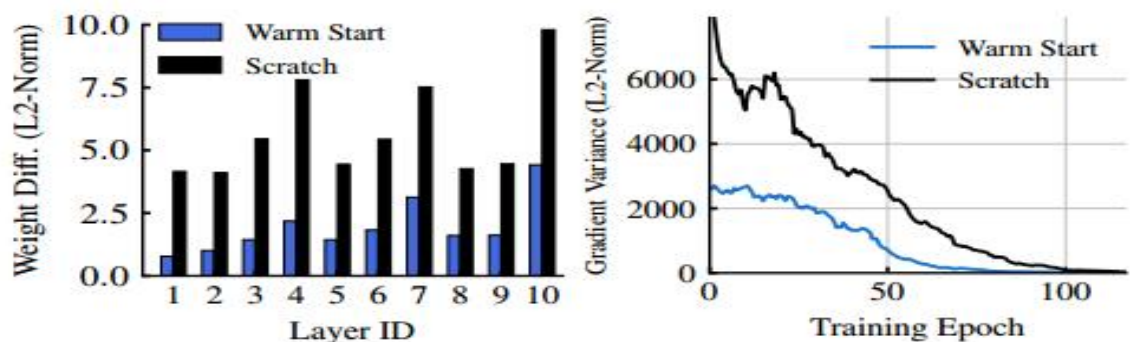
动机二：相似的模型可以预热模型训练

与通常关注在不同数据集上手动传输相同模型以提高准确性不同，作者注意到，将已训练模型的权重转换到新模型可以加速其训练过程。从 figure3 和 figure4 中可以看出，相比于从头开始训练模型，将不同模型结构的权重复制到当前查询模型中可以实现准确率基本没有太大差异的基础上加速模型的收敛。特别是，figure3 (b) 展示出了模型的收敛速度不仅收到父模型准确率的影响，同时受到父模型与查询模型相似度的影响。



(a) Warm start accelerates training. (b) Parent model accuracy matters.

Figure 3: Transferring model weights from well-trained models with similar architectures can accelerate new model training.



(a) Smaller divergence to the optimal. (b) Smaller gradient variance.

Figure 4: Warm start provides better initial weights search space. We use RoR3 to warm start ResNet101.

核心贡献

1. 自动化预热系统: ModelKeeper 是第一个为深度神经网络 (DNN) 提供自动训练预热的系统, 通过使用相似模型的权重来初始化新的训练任务。
2. 模型相似性匹配: 该论文介绍了一种两阶段分层匹配系统, 用于识别和选择最合适的“父”模型, 在权重初始化时考虑相似性和准确性。
3. 转换技术: ModelKeeper 应用结构感知转换技术, 包括宽度和深度运算符, 在不同架构之间重新利用权重, 最大化信息保留而不会产生显著的额外开销。
4. 集成与可扩展性: 该系统与流行的机器学习框架 (如 Ray、AutoKeras 和 PyTorch) 集成, 展示了在数千个模型中灵活性和可扩展性。

方法论和设计

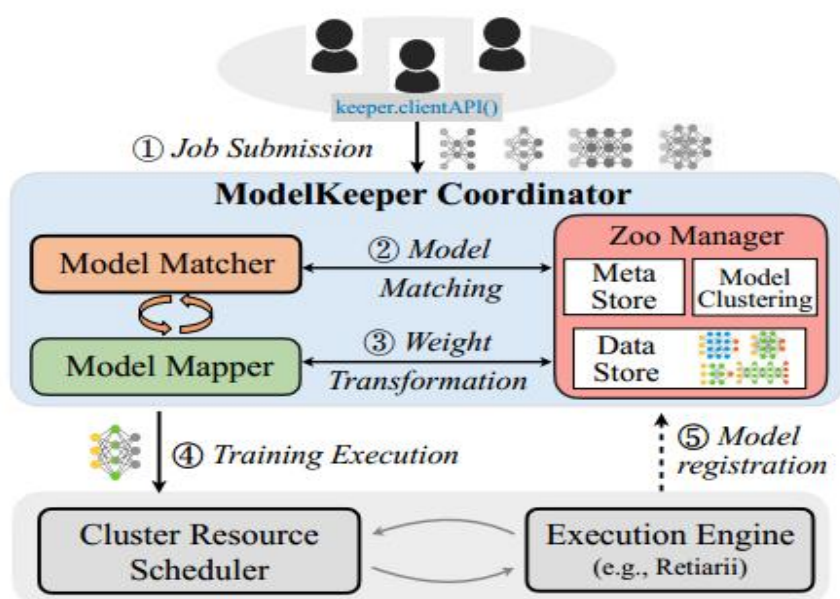


Figure 5: ModelKeeper architecture. It can run as a cluster-wide service to serve different users and/or frameworks.

1. 架构：ModelKeeper 的架构包括一个中央协调器和客户端代理。协调器负责模型匹配和权重转换，而客户端处理用户请求和预热初始化。
2. 模型相似性评估：模型相似性通过基于动态规划的方法来衡量，从图层级别捕捉架构的相似性。
3. 权重转换：为了最大化可重用权重的数量，ModelKeeper 使用适应模型维度的运算符，允许部分匹配。此方法优于仅考虑相同层的传统方法。

当开发人员创建新的训练任务时，①首先建立与远程 ModelKeeper 协调器的客户端连接，并通过查询提交指定的任务元信息。ModelKeeper 客户端代理会自动提取所需的模型信息（如模型计算图），并向协调器发送请求（请求大小通常小于 1 MB）。②一旦接收到请求，匹配器会查阅其元数据存储，识别用户可访问并满足指定标签（如首选父模型名称）的模型库模型，并测量它们与查询模型的架构相似性。③映射器从这些模型库模型中选择一个具有较大架构相似性和良好准确性的父模型。随后，它从数据存储中加载该父模型的权重，并根据匹配器的成对张量映射，将父模型的权重转换为查询模型的格式。请注意，此转换仅

更新张量权重值，而其他内容（如模型架构）保持不变。④协调器将预热后的模型权重返回给开发人员，之后的训练过程按常规进行。⑤训练完成后，ModelKeeper 可以自动将已训练的模型注册到模型库管理器，以便未来任务使用。

```

1 from modelkeeper import ModelKeeperClient
2
3 def training_with_keeper(model, dataset):
4     # Create client session to keeper coordinator
5     keeper_client = ModelKeeperClient(coordinator_ip)
6     warmed_model, meta = keeper_client.query_for_model(
7         model, meta={'data': 'Flowers102',
8                     'task': 'classification', 'tags': None})
9
10    acc = train(warmed_model, dataset) # Training starts
11
12    # Register model to ModelKeeper when training ends
13    keeper_client.register_model(warmed_model,
14                                meta={'data': 'Flowers102', 'accuracy': acc,
15                                    'task': 'classification', 'tags': None})
16    keeper_client.stop()

```

Figure 6: Code snippet of ModelKeeper client service APIs.

①Model Matcher

使用基于类动态规划启发式的结构感知成对模型匹配机制进行模型之间相似度的计算：该算法按图张量的拓扑顺序遍历相似性表。这使我们能够嵌入图层级信息，同时优先匹配前缀。为了推进到当前的张量对 (i, j) ，算法枚举了三种可能的操作：

1. 匹配（MATCH）：将 i 的父节点的权重转换为 j 的父节点。
2. 跳过父节点（SKIP_PARENT）：放弃转换张量 i 的父节点；
3. 跳过子节点（SKIP_CHILD）：放弃转换到张量 j 的父节点；

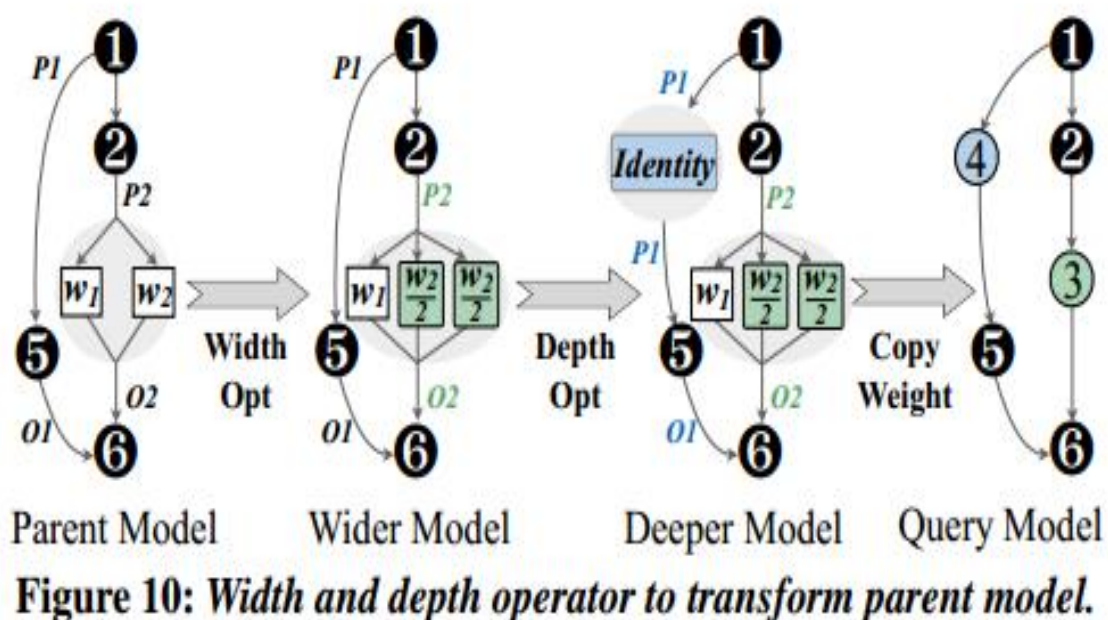
$$M(i, j) = \max_{k \in \text{parent}(i)} \begin{cases} M(k, j_{\text{parent}}) + \text{MATCH}(k, j_{\text{parent}}) & (1) \\ M(k, j) + \text{SKIP_PARENT} & (2) \\ M(i, j_{\text{parent}}) + \text{SKIP_CHILD} & (3) \end{cases}$$

②Model Mapper

1. 基于相似性和准确性选择最佳父模型。

映射器采用流行的分桶策略，根据相似性将模型分配到 B 个桶中。在默认情况下 $B = 10$ ，第 10 个桶将容纳相似度在 0.9 到 1.0 之间的模型，因此同一个桶中的模型具有相近的相似性。然后，映射器从最后一个桶（第 10 个桶）向第一个桶遍历，直到找到一个包含模型的桶（第 9 个桶），并从中选择性能最佳的模型作为父模型。这样一来，父模型在模型相似性和准确性方面接近最佳边界。

2. 使用宽度和深度转换运算符来调整权重。



3. 部分映射保留来自非相同架构的信息。

扩展情况：

从宽度角度：为了将父模型张量转换为更宽的查询模型张量，宽度运算符将父模型权重复制到查询模型的映射张量，并通过其他列的加权复制来填充其余列。

从深度角度：当映射需要在父模型中插入一个新张量（即 `SKIP_CHILD`）时，深度运算符会将此映射张量的权重初始化为单位张量。

剪枝情况：

从宽度角度：当需要将较宽的张量权重适配到查询模型的较小维度张量时，宽度运算符将逐步选择并复制父模型张量中最大的权重值到查询模型的映射张量。

从深度角度：当我们跳过转换父模型的张量（即 `SKIP_PARENT`）时，深度运算符会向该张量的邻近权重值添加噪声。

③ Zoo Manager

- 1.两阶段分层模型匹配：分层聚类模型，以高效处理大规模预热。
- 2.限制模型库规模：使用基于背包的模型保留方法来管理存储限制。
- 3.避免低准确度模型：筛选低准确度模型，以保持高质量的预热资源。

评估

作者在五个模型库中对 `ModelKeeper` 进行了评估，涵盖了不同的计算机视觉（CV）和自然语言处理（NLP）任务，并将其性能与 `Retiarii` 和 `AutoKeras` 等其他先进系统进行了比较。结果显示：

1. 训练加速：`ModelKeeper` 在各种任务中实现了 1.3 倍到 4.3 倍的训练加速，节省了 23%-77% 的训练时间。
2. 跨模型一致性：即使在不利环境（如模型库中存在低准确度模型）下也表现良好。
3. 准确性：由于更加智能的权重初始化，`ModelKeeper` 能保持或略微提升模型的准确性。

Category	Task	Workload	# of Models	Dataset	Avg. Time Improvement	Avg. Acc. Difference
Exploratory Training	Grid Search NAS	NASBench [24]	1,000	CIFAR-100	2.9×	0.39%
	Evolution NAS				2.4×	0.38%
	AK-Bayesian NAS [41]	AutoKeras Zoo [41]	500		4.3×	0.31%
General Training	Image Classification	Imgclsmb [10]	389	Flowers102 [54]	2.8×	0.23%
		Imgclsmb-Small	179	CIFAR-10	2.1×	0.02%
				CIFAR-100	1.6×	0.18%
				ImageNet32 [19]	1.3×	0.03%
	Ensemble Training	V-Ensemble [65]	104	CIFAR-100	1.7×	0.08%
	Language Modeling	HuggingFace [2]	69	WikiText-103 [47]	1.8×	-0.13 perplexity

Table 1: Summary of improvements. `ModelKeeper` improves training execution time without accuracy drop, by reducing the amount of training needed (i.e., GPU Saving). Accuracy difference is defined by $Acc.(Keeper) - Acc.(Baseline)$, and smaller perplexity is better.

不足和未来的工作

论文指出了几个改进方向：

1. 支持超参数调优: **ModelKeeper** 目前在预热过程中未考虑优化超参数。集成自适应学习率可能会进一步提高训练速度。
2. 跨数据集的泛化: 尽管在相似数据集中效果良好, 但在差异较大的任务或数据集之间的预热性能仍是一个未解决的问题。
3. 集群环境中的模型共享: 为实现更广泛的应用, 论文建议探索共享库来存储适合预热的模型, 这可能有助于实现跨组织的模型复用。

结论

1. **ModelKeeper** 实现了高效的深度神经网络训练自动预热。
2. 在保持准确性的同时, 显著节省时间。
3. 为拥有大型模型库的共享机器学习集群提供了可扩展的解决方案。